

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [??????](#) • [Svenska](#) • [???\(????\)?](#) • [???\(??\)?](#) •

Contents

- [1 Introduction](#)
 - ◆ [1.1 Why Transmission?](#)
 - ◆ [1.2 Why DD-WRT + Transmission?](#)
- [2 Instructions](#)
 - ◆ [2.1 Prerequisites](#)
 - ◆ [2.2 Starting up](#)
 - ◆ [2.3 Transmission Config Directory](#)
 - ◆ [2.4 Transmission Download Directory](#)
 - ◆ [2.5 Whitelist IP's](#)
 - ◆ [2.6 Web UI Port](#)
 - ◆ [2.7 Web UI Style](#)
 - ◆ [2.8 Username and Password](#)
 - ◆ [2.9 Troubleshooting](#)
- [3 Reference](#)

Introduction

Why Transmission?

Rather than downloading a file from a single source server, the BitTorrent protocol allows users to join a "swarm" of hosts to download and upload from each other simultaneously. This way impact of distributing large files are reduced to the network and server. [Transmission \(BitTorrent client\)](#) is a BitTorrent client which features a variety of user interfaces on top of a cross-platform back-end. It includes a built-in web server so that users can control Transmission remotely via the web. Within the file selection menus, users can customize their downloads at the level of individual files. Transmission allows the assigning of priorities to torrents and to files within torrents, thus potentially influencing which files are being downloaded first. It supports the Magnet URI scheme and encrypted connections. It allows torrent-file creation and peer exchange. It also supports automatic port-mapping using UPnP/NAT-PMP, peer caching, blocklists for bad peers, bandwidth limits dependent on time-of-day, globally or per-torrent, and has partial support for IPv6. It allows the use of multiple trackers simultaneously, Local Peer Discovery, Micro Transport Protocol (µTP), and UDP tracker.

Why DD-WRT + Transmission?

DD-WRT builds starting from March 2019 have new and modern way of downloading and managing torrents with transmission-daemon. Everything is done through GUI in just a few clicks, no need for optware installation or Entware.

BrainSlayer, Sebastian Gottschall, lead dd-wrt developer, configured it to work out of the box but in a secure manner (remote web access requires authentication and only whitelist IP's entered by user are allowed).

- Note: Entware or Optware is required for transmission-daemon for small (<4MB) dd-wrt builds, as transmission-daemon is not built-in in those builds.

Instructions

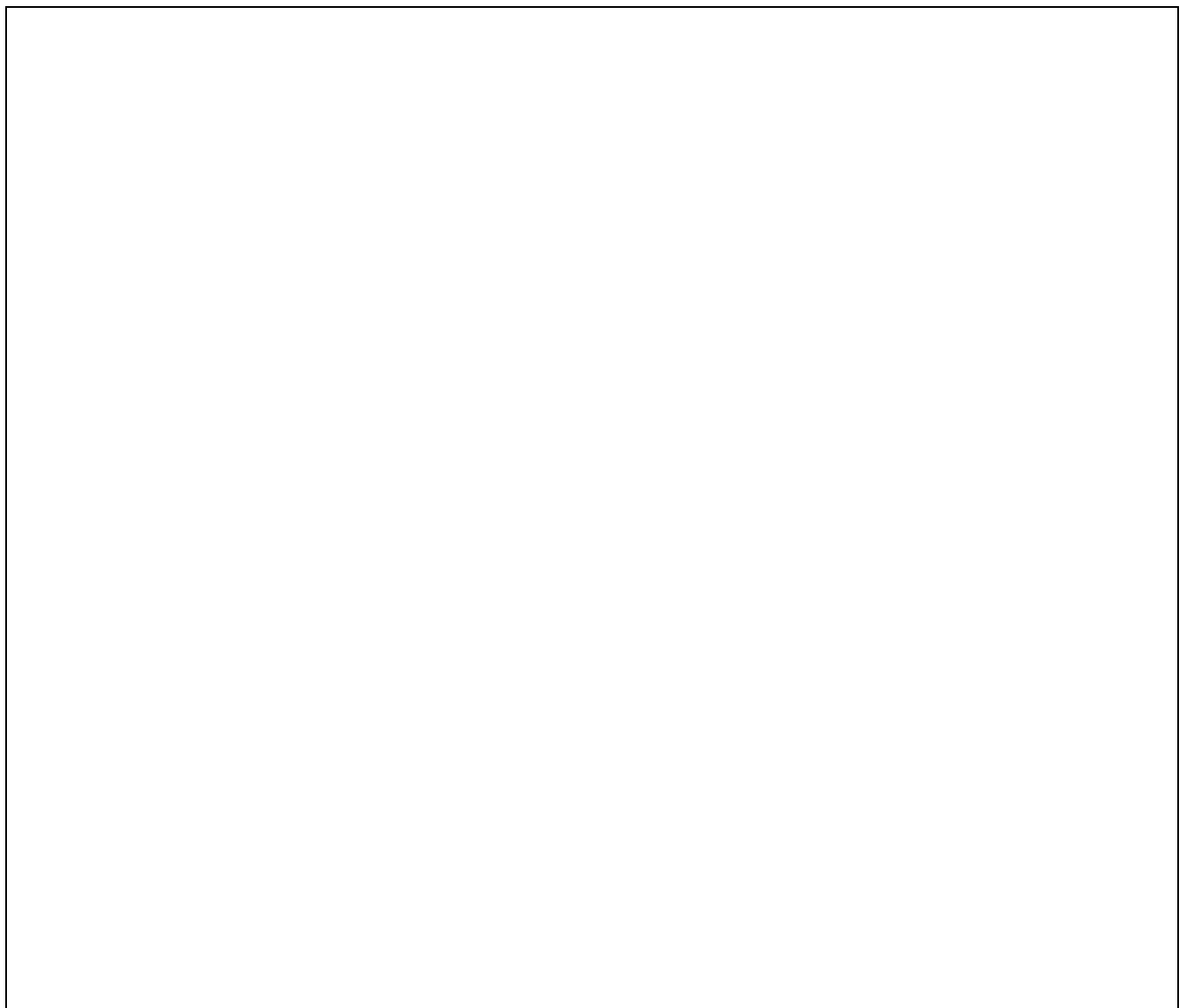
Prerequisites

We'll assume that you have enabled Core USB Support/Storage Support, formatted NAS device/USB stick and mounted it as /mnt. Also, you could enable SAMBA if you want your NAS to be accessible through localnet. Create users and give permissions. Note: to access samba share on linux (Nemo, Nautilus, Caja) you can use smb://username@router-ip.



Starting up

transmission-daemon will start automatically when you enable it from GUI and click apply, you just need to fill few parameters.



Transmission Config Directory

This is the path where settings.json is stored with defined settings. No need to changing it manually unless you know what you are doing. /mnt/sda/transmission_config

Transmission Download Directory

This is the path where downloaded file are stored. /mnt/sda/

Whitelist IP's

rpc-whitelist defines access to transmission. Localhost (127.0.0.1) and local subnet (192.168.1.*) is defined by

default. If you use Wireguard or OpenVPN to access your localnet you can enter here IP (e.g. 172.16.10.*) subnet of tunneled interface to enable remote web access to transmission.

Web UI Port

you can change it but notice that this will change default web access too...

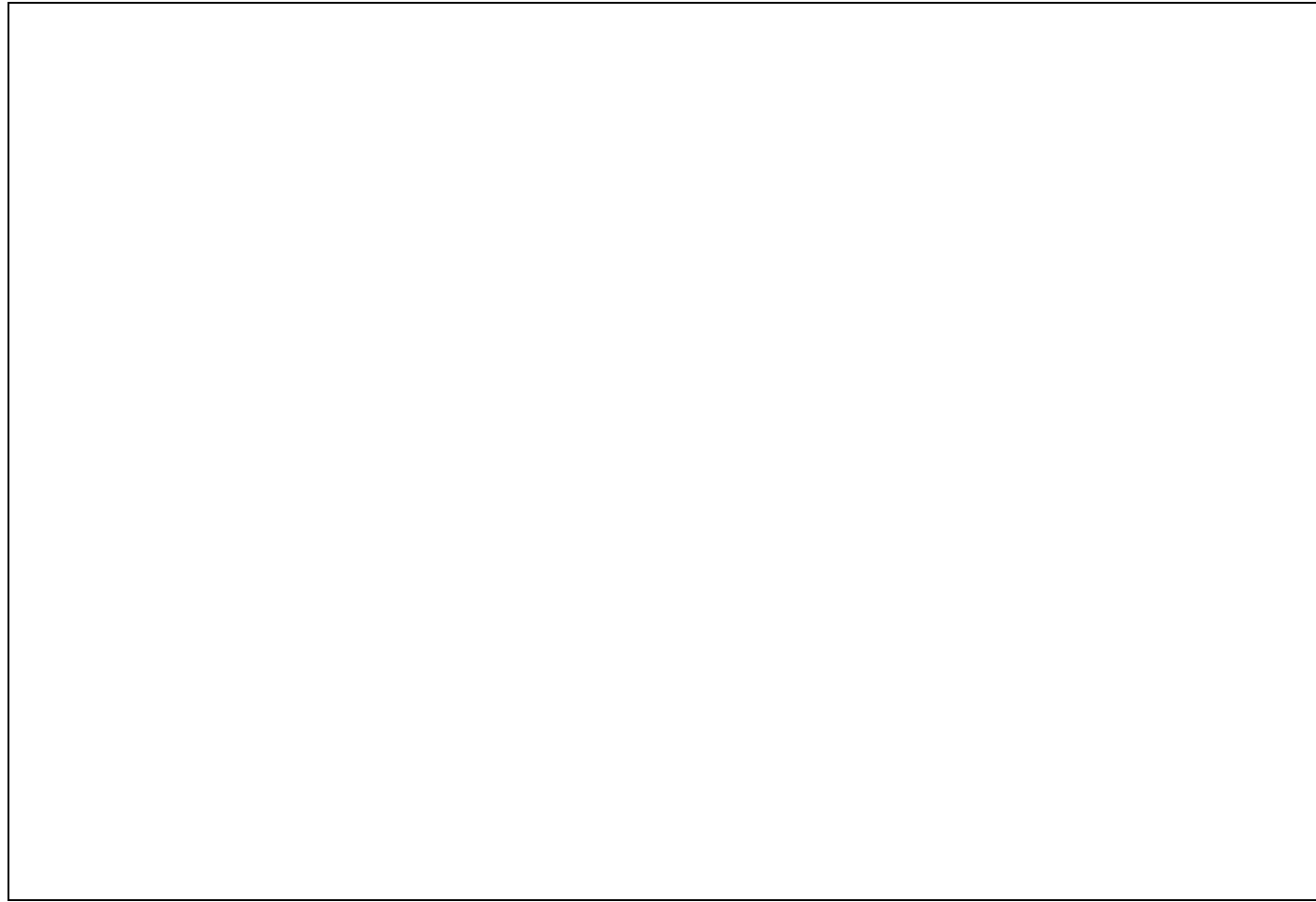
<http://server-ip:9091>

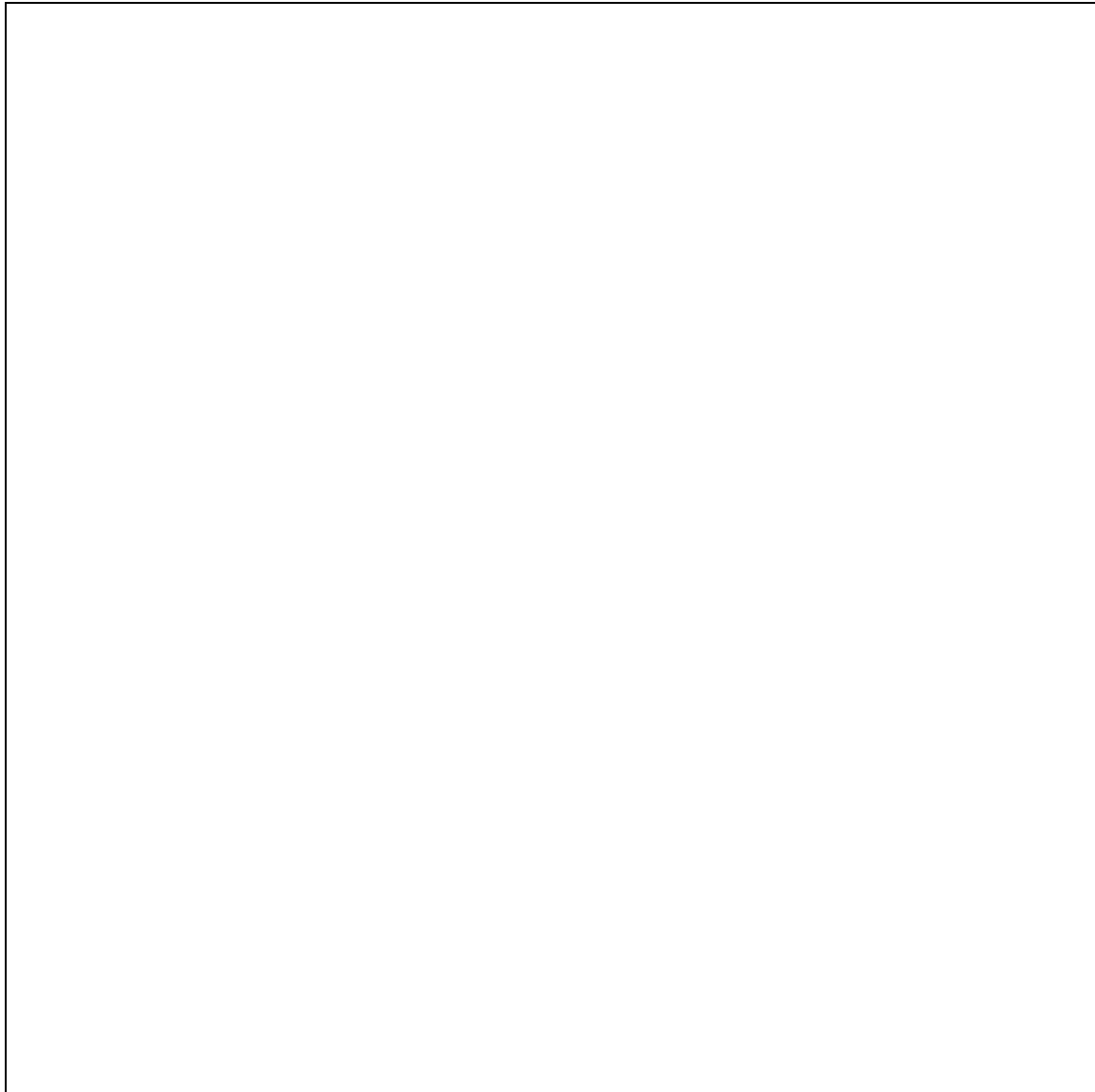
Web UI Style

Combustion is the one I prefer.

Username and Password

Change it to whatever you want (any password will work). After next restart the password will be rewritten in SHA1 encrypted format for security reasons.





Troubleshooting

Successful installation of transmission-daemon should export such in your log:

```
transmission-daemon: Transmission 2.94 (d8e60ee44f) started (session.c:740)
transmission-daemon: RPC Server Adding address to whitelist: 127.0.0.1 (rpc-server.c:971)
transmission-daemon: RPC Server Adding address to whitelist: 192.168.2.* (rpc-server.c:971)
transmission-daemon: RPC Server Adding address to whitelist: 172.168.10.* (rpc-server.c:971)
transmission-daemon: RPC Server Serving RPC and Web requests on port 127.0.0.1:9091/transmission/
transmission-daemon: RPC Server Whitelist enabled (rpc-server.c:1217)
transmission-daemon: RPC Server Password required (rpc-server.c:1220)
transmission-daemon: DHT Reusing old id (tr-dht.c:307)
transmission-daemon: DHT Bootstrapping from 99 IPv4 nodes (tr-dht.c:156)
transmission-daemon: Port Forwarding (NAT-PMP) initnatpmp succeeded (0) (natpmp.c:70)
transmission-daemon: Port Forwarding (NAT-PMP) sendpublicaddressrequest succeeded (2) (natpmp.c:7
transmission-daemon: Using settings from "/mnt/sda/transmission_config" (daemon.c:528)
transmission-daemon: Saved "/tmp/mnt/sda/transmission_config/settings.json" (variant.c:1266)
transmission-daemon: transmission-daemon requiring authentication (daemon.c:554)
```

Easy_torrenting_with_dd-wrt

```
transmission-daemon: Port Forwarding State changed from "Not forwarded" to "Starting" (port-forwa  
transmission-daemon: Loaded 1 torrents (session.c:2034)  
transmission-daemon: archlinux-2019.02.01-x86_64.iso Starting IPv4 DHT announce (good, 172 nodes)  
transmission-daemon: archlinux-2019.02.01-x86_64.iso IPv4 DHT announce done (tr-dht.c:545)
```

Reference

[Conceptual Overview](#)

[Git Repository](#)

[DD-WRT SVN Changesets](#)

[Install Optware Transmission](#)